

Adversarial Score Distillation: When score distillation meets GAN

Min Wei^{1*} Jingkai Zhou^{2*} Junyao Sun³ Xuesong Zhang¹

¹Beijing University of Posts and Telecommunications ²Independent Researcher ³South China University of Technology
{mw, xuesong_zhang}@bupt.edu.cn fs.jingkaizhou@gmail.com 201810108390@mail.scut.edu.cn

Abstract

Existing score distillation methods are sensitive to classifier-free guidance (CFG) scale, manifested as over-smoothness or instability at small CFG scales, while oversaturation at large ones. To explain and analyze these issues, we revisit the derivation of Score Distillation Sampling (SDS) and decipher existing score distillation with the Wasserstein Generative Adversarial Network (WGAN) paradigm. With the WGAN paradigm, we find that existing score distillation either employs a fixed sub-optimal discriminator or conducts incomplete discriminator optimization, resulting in the scale-sensitive issue. We propose the Adversarial Score Distillation (ASD), which maintains an optimizable discriminator and updates it using the complete optimization objective. Experiments show that the proposed ASD performs favorably in 2D distillation and text-to-3D tasks against existing methods. Furthermore, to explore the generalization ability of our paradigm, we extend ASD to the image editing task, which achieves competitive results. The project page and code are at [this link](#).

1. Introduction

Score distillation is a rapidly growing technique attracting a lot of attention [17, 27, 36, 44, 55, 56, 62]. It transfers knowledge from a pretrained 2D diffusion model to downstream tasks, such as image editing [17, 27], video editing [27], diffusion model distillation [36, 47], and text-to-3D [4, 7, 8, 32, 39, 44, 48, 55, 56, 61, 62], with no downstream data required.

However, as a milestone in score distillation methods, Score Distillation Sampling (SDS) [44] is very sensitive to the classifier-free guidance (CFG) [18] scale. Figure 1 takes 2D score distillation as an example. When setting a small CFG scale, SDS tends to obtain over-smoothing results, whereas when setting a large one, the generated images become over-saturated. Recently, Variational Score Distillation (VSD) [56] alleviates the over-smoothing problem at



(a) SDS is sensitive to the CFG scale.



Figure 1. 2D score distillation examples with the prompts “a photograph of an astronaut riding a horse” and “exterior frontal perspective shot of resort villa inspired by Mykonos architecture”. SDS is very sensitive to the CFG scales while VSD exhibits fluctuation of generated contents during distillation at small CFG scales.

small CFG scales, but we observe that VSD is a bit unstable during the distillation progress at small CFG scales, as shown in Figure 1. These phenomena make us conjecture that there are still some unrevealed methodological issues behind SDS and VSD.

We start from revisiting the derivation of SDS. In [44], the gradient of SDS is derived from the L2 loss of the diffusion model. This derivation holds only when the vanilla-predicted noise is used in the gradient. However, in practice, the SDS gradient exploits the CFG noise rather than the vanilla-predicted one. This little trick changes the whole thing. It means that, in practice, the real SDS loss is no longer the L2 loss of the diffusion model, but some other loss associated with the implicit classifier or discriminator.

*Equal contribution

Corresponding authors: Jingkai Zhou, Xuesong Zhang



a dog with its reflection below

*a DSLR photo of a hamburger
inside a restaurant*

*a professional photo of a sunset
behind the Grand Canyon*

a dumpster full of trash

(a) ASD can generate photorealistic images through 2D score distillation.



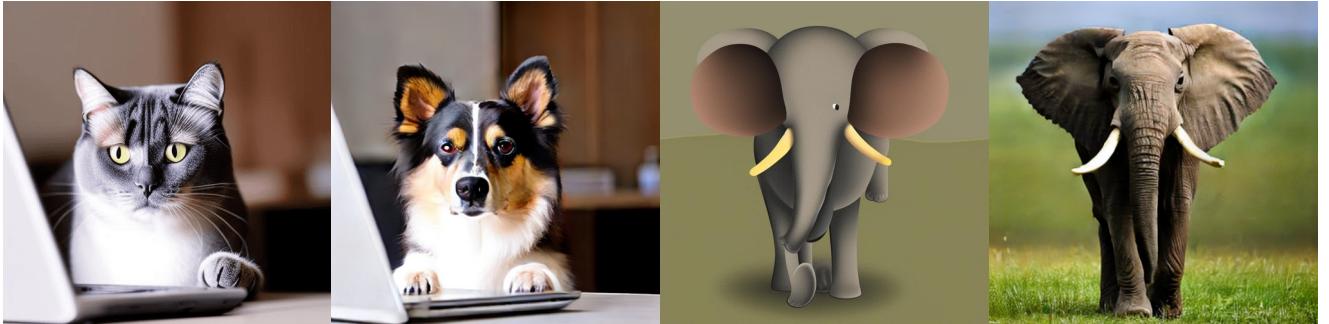
*a freshly baked loaf of sourdough
bread on a cutting board*

a ficus planted in a pot

a roast turkey on a platter

a pineapple

(b) ASD can generate high quality 3D NeRFs from scratch (only stage 1 of VSD [56]).



*a cat is posing next to
a laptop computer*

*a dog is posing next to
a laptop computer*

A cartoon elephant

A cartoon elephant

(c) ASD can extend to image editing with caption modification.

Figure 2. Examples generated by ASD in 2D distillation, image editing, and text-to-3D tasks. Stable Diffusion [51, 52] is used as the pretrained diffusion model. For more results please refer to our project page.

To figure out where the real SDS gradient comes from, we connect SDS to Generative Adversarial Networks (GANs) [3, 13, 58]. By carefully designing the discriminator using the diffusion model, we can naturally derive the SDS gradient from the generator loss in the Wasserstein GAN (WGAN) [3, 58]. This means that SDS inherently comes from WGAN. However, when we employ the WGAN paradigm to explain and analyze SDS, we find

that the discriminator optimization in WGAN is ignored by SDS. More specifically, SDS only optimizes the generator loss in WGAN by using a fixed sub-optimal discriminator implemented with added noise. Similarly, VSD can also be represented with the WGAN paradigm, in which case the discriminator is optimizable but the optimization objective is incomplete compared to the WGAN discriminator loss, making the distillation progress unstable.

In this work, we propose Adversarial Score Distillation (ASD) based on the WGAN paradigm. ASD maintains an optimizable discriminator and optimizes it using the complete WGAN discriminator loss, thereby improving distillation stability and quality. Specifically, the discriminator can be implemented using the combination of diffusion models, where textual-inversion embedding [12] or LoRA [20] make it optimizable. During discriminator optimization, we propose two kinds of optimization objectives derived from the complete WGAN discriminator loss. One exploits both real sample distribution and a pretrained diffusion model and can therefore be applied to the image-conditioned task, such as image editing. The other only requires a pretrained diffusion model, which is suitable for text-conditioned cases. Experiments on 2D distillation and text-to-3D tasks demonstrate the superior performance of ASD compared to existing score distillation methods. See Figure 2 for some examples. In addition, we extend ASD to image editing via discriminator transformations and show that the Delta Denoising Score (DDS) [17] is a special case in our paradigm. Figure 2 and experiments show some competitive editing results. This demonstrates that our paradigm can be generalized to more GAN paradigms for various tasks.

Beyond methodological improvements, the more important significance of bridging score distillation and GANs is to enable GAN paradigms to exploit powerful diffusion models in the form of score distillation. More specifically, we can extend a general diffusion model to various downstream tasks (*e.g.*, text-to-3D, image editing, diffusion distillation) by designing GAN paradigms, which eliminates the need to redesign specific diffusion models or collect large amounts of finetuning samples.

In short, we make the following contributions:

- Bridged score distillation and WGAN to explain the methodological issues of existing score distillation, which also enables the pretrained model to extend to various downstream tasks by designing GAN paradigms.
- Proposed ASD based on the WGAN paradigm that employs the complete WGAN discriminator loss, resulting in better distillation stability and quality.
- Comprehensive experiments show that the proposed ASD performs favorably against existing methods in 2D distillation, text-to-3D, and image editing tasks.

2. Related Work

Score Distillation. Score distillation, first proposed by [44] and [55], is a method to optimize the parameter space via distilling a pretrained diffusion model. In text-to-3D generation, Score Distillation Sampling (SDS) shows great potential for optimizing 3D representations based on differentiable rendering. However, SDS often suffers from problems such as over-saturation, over-smoothing, and low

diversity. The recent work Variational Score Distillation (VSD) [56] has made tremendous progress in solving these problems, which optimizes a 3D distribution by employing the Wasserstein gradient flow. However, VSD is a bit unstable during the optimization progress at small CFG scales, which will be discussed later. Score distillation can also be used in image editing [17, 50]. Delta Denoising Score (DDS) [17] proposes to modify the image in a zero-shot way by only editing its caption. Moreover, StyleGAN-Fusion [50] and Diff-Instruct [36] apply score distillation to knowledge distillation between generators in a data-free manner. Our work will trace the origin of SDS and connect it with GAN to reveal its methodological issues.

Generative Adversarial Networks. GANs [2, 3, 13, 14, 24, 38, 58] have achieved great success in generating realistic and clear images, such as image in-painting [10, 22], image manipulation applications [5, 6, 16, 42, 64], image-to-image translation [9, 21, 23, 28, 30, 34, 54, 60, 63]. Among these 2D GANs, Wasserstein GAN (WGAN) [3] is a remarkable study that minimizes the Wasserstein distance instead of the Jensen-Shannon divergence to better handle the case when the generated sample distribution is far away from the real distribution. In 3D modeling, a lot of work on 3D GANs [1, 49, 57, 59] has been proposed. However, the generation of 3D representations from low-dimensional latent spaces needs expensive high-quality 3D data for training. GAN2Shape [43] demonstrates that conventional 2D GANs, when exclusively trained on images, contain rich 3D knowledge and can faithfully reconstruct intricate 3D shapes without 3D annotations. In this work, we only use the form of WGAN to explain and analyze score distillation methods, such as SDS and VSD.

3. Preliminaries

Score Distillation Sampling. Score Distillation Sampling (SDS) [44] proposes to use 2D diffusion priors to optimize the parameters θ of 3D representations. It first employs a differentiable renderer $g(\theta, c)$ parameterized by θ to generate an image $x_0^g = g(\theta, c)$ at a random view c , then adds noise to this image x_0^g to obtain x_t^g , and uses a pretrained diffusion model to predict noise conditioned on a given text y . In the SDS paper [44], the gradient is directly derived from the L2 loss of the diffusion model, which can be written as

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\text{diff}} &= \nabla_{\theta} \mathbb{E}_{t, \epsilon} [\omega(t) \| \epsilon_{x_t^g; y, t} - \epsilon \|_2^2] \\ &= \mathbb{E}_{t, \epsilon} [\omega(t) (\epsilon_{x_t^g; y, t} - \epsilon) \frac{\partial \epsilon_{x_t^g; y, t}}{\partial x_t^g} \frac{\partial x_t^g}{\partial x_0^g} \frac{\partial x_0^g}{\partial \theta}] \\ &\approx \mathbb{E}_{t, \epsilon} [\omega(t) (\epsilon_{x_t^g; y, t} - \epsilon) \frac{\partial x_0^g}{\partial \theta}] = \nabla_{\theta} \mathcal{L}_{\text{SDS}} \end{aligned} \quad (1)$$

where $\omega(t)$ is a weight based on the time t , and $\epsilon_{x_t^g; y, t}$ is the vanilla-predicted noise with input x_t^g conditioned on y

and t . However, in practice, SDS uses the classifier-free guidance (CFG) [18] noise, that is

$$\tilde{\epsilon}_{x_t^g; y, t} = \epsilon_{x_t^g; t} + \lambda(\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; t}) \quad (2)$$

where $\epsilon_{x_t^g; t}$ is the vanilla-predicted noise with input x_t^g conditioned on t , and λ is the CFG scale. Replacing $\epsilon_{x_t^g; y, t}$ with $\tilde{\epsilon}_{x_t^g; y, t}$ makes the derivation of Eq. 1 no longer valid. In other words, $\nabla_\theta \mathcal{L}_{\text{SDS}}$ should be written as

$$\nabla_\theta \mathcal{L}_{\text{SDS}} = \mathbb{E}_{t, \epsilon}[\omega(t)(\epsilon_{x_t^g; t} + \underbrace{\lambda(\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; t})}_{\text{grad from implicit classifier}} - \epsilon) \frac{\partial x_0}{\partial \theta}] \quad (3)$$

which cannot be derived from $\mathcal{L}_{\text{diff}}$, but rather from another loss related to the implicit classifier or discriminator.

Generative Adversarial Networks. Generative Adversarial Networks (GANs), first introduced in [13], can be formulated by a zero-sum competition between a discriminator network D and a generator network G . The original form of GAN [13] essentially minimizes the Jensen-Shannon (JS) divergence between the distributions of real samples and generated samples, which gets a gradient close to zero when two distributions are very different. The Wasserstein GAN (WGAN) [3, 58] reformulates the original GAN with the Wasserstein distance to handle the zero-gradient issue, which can be written as

$$\min_G \max_D \mathbb{E}_{\mu_r}[D(x^r)] - \mathbb{E}_{\mu_g}[D(x^g)] - \underbrace{\tau \mathbb{E}_{\tilde{\mu}}[\|\nabla_{\tilde{x}} D(\tilde{x})\|^p]}_{\text{penalty term}} \quad (4)$$

where x^r comes from the distribution μ_r of real samples, x^g comes from the distribution μ_g of generated samples, \tilde{x} comes from a mixture distribution $\tilde{\mu}$ of first two, $\|\cdot\|^p$ denotes p-norm, and $\tau \geq 0$ is the penalty weight.

4. Adversarial Score Distillation

One goal of this work is to unravel the methodological issues behind SDS by revisiting and discovering the origin of the SDS gradient. As the gradient used in practice should originate from the loss associated with the implicit classifier or discriminator, it is natural to connect SDS with GAN. After carefully building a discriminator using the diffusion model, we can derive the SDS gradient directly from the generator loss in the WGAN.

We define a discriminator in the form as

$$D(x_t; y) = \log \frac{p(y|x_t)}{p(\phi|x_t)} \quad (5)$$

where $p(y|x_t)$ denotes the probability of the given noisy sample $x_t = \alpha_t x_0 + \sigma_t \epsilon$ to be categorized as the samples described by prompt y , $p(\phi|x_t)$ represents the probability of x_t to be categorized as the samples described by prompt ϕ . Here, we consider ϕ as the prompt for fake samples.

When x_t is close to samples described by prompt y , this $D(x_t; y)$ will give high confidence. Otherwise, it will get low confidence when x_t is close to samples described by prompt ϕ . Based on WGAN [3, 58], we have the generator loss as

$$\begin{aligned} \mathcal{L}_G &= \mathbb{E}_{t, \epsilon}[-D(x_t^g; y)] = \mathbb{E}_{t, \epsilon}[-\log \frac{p(y|x_t^g)}{p(\phi|x_t^g)}] \quad (6) \\ &\leq \mathbb{E}_{t, \epsilon}[-\log \frac{p(y|x_t^g)^\lambda}{p(\phi|x_t^g)}] := \mathcal{L}'_G \quad s.t. \lambda \geq 1 \end{aligned}$$

where $x_t^g = \alpha_t g(\theta, c) + \sigma_t \epsilon$ denotes the generated sample with noise. According to the Bayesian rule, we can do the transformation $\nabla_{x_t} \log p(y|x_t) = -1/\sigma_t (\epsilon_{x_t; y, t} - \epsilon_{x_t, t})$ similar to CFG [18]. Thus, we have

$$\begin{aligned} \nabla_\theta \mathcal{L}'_G &= \nabla_\theta \mathbb{E}_{t, \epsilon}[\log p(\phi|x_t^g) - \lambda \log p(y|x_t^g)] \quad (7) \\ &= \mathbb{E}_{t, \epsilon}[\omega(t)(\epsilon_{x_t^g; t} + \lambda(\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; t}) - \epsilon_{x_t^g; \phi, t}) \frac{\partial x_0}{\partial \theta}] \end{aligned}$$

where $\omega(t)$ is a weight based on the time t . Compare the SDS gradient in Eq. (3) with the Eq. (7), the only difference is that the SDS gradient uses the added noise ϵ to approximate the noise $\epsilon_{x_t^g; \phi, t}$. This approximation may not be optimal, because $\epsilon_{x_t^g; \phi, t}$ should come from the diffusion model, so that $\epsilon_{x_t^g; \phi, t} - \epsilon_{x_t^g; t}$ approximates $-\sigma_t \nabla_{x_t} \log p(\phi|x_t)$. From another intuitive perspective, $\epsilon_{x_t^g; \phi, t}$ contains the inductive bias of the pretrained diffusion model while ϵ does not. Recently, VSD [56] uses the LoRA branch to predict $\epsilon_{x_t^g; \phi, t}$ which also contains the inductive bias of the pretrained diffusion model, thereby achieving better quality.

Let us go further into discriminator optimization. We have the discriminator loss, based on Eq. (4) as

$$\begin{aligned} \mathcal{L}_D &= \mathbb{E}_{t, \epsilon}[D(x_t^g; y) - D(x_t^r; y) + \tau \|\nabla_{\tilde{x}_t} D(\tilde{x}_t)\|_2^2] \quad (8) \\ &= \mathbb{E}_{t, \epsilon}[\log \frac{p(y|x_t^g)}{p(\phi|x_t^g)} - \log \frac{p(y|x_t^r)}{p(\phi|x_t^r)} + \tau \|\nabla_{\tilde{x}_t} \log \frac{p(y|\tilde{x}_t)}{p(\phi|\tilde{x}_t)}\|_2^2] \end{aligned}$$

where $x_t^r = \alpha_t x_0^r + \sigma_t \epsilon$ is the noisy real sample based on the real sample x_0^r and the time t . In Eq. (8), $p(y|x_t^r)$ and $p(y|x_t^g)$ are based on the given pretrained diffusion model, so they are not optimizable. ϕ in $p(\phi|x_t^g)$ and $p(\phi|x_t^r)$ is optimizable, which can be implemented using an textual-inversion embedding [12] or LoRA [20]. Thus, \mathcal{L}_D is a loss function only about ϕ , its gradient can be derived as

$$\begin{aligned} \nabla_\phi \mathcal{L}_D &= \nabla_\phi \mathbb{E}_{t, \epsilon}[\log \frac{p(\phi|x_t^r)}{p(\phi|x_t^g)} + \tau \|\nabla_{\tilde{x}_t} \log \frac{p(y|\tilde{x}_t)}{p(\phi|\tilde{x}_t)}\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t, \epsilon}[\log \frac{p(x_t^r|\phi)}{p(x_t^g|\phi)} + \tau \|\nabla_{\tilde{x}_t} \log \frac{p(\tilde{x}_t|y)}{p(\tilde{x}_t|\phi)}\|_2^2] \quad (9) \\ &\approx \nabla_\phi \mathbb{E}_{t, \epsilon}[\|\epsilon_{x_t^g; \phi, t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r; \phi, t} - \epsilon\|_2^2 \\ &\quad + \underbrace{\eta \|\epsilon_{x_t^r; y, t} - \epsilon_{x_t^r; \phi, t}\|_2^2 + \gamma \|\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; \phi, t}\|_2^2}_{\text{penalty term}}] \end{aligned}$$

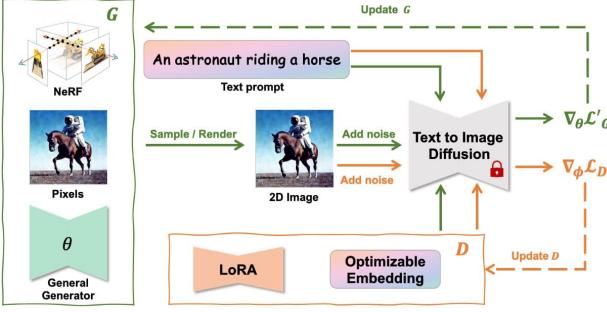


Figure 3. Workflow of ASD. Green lines show the pipeline of generator optimization. Orange lines show the pipeline of discriminator optimization. The avatar of NeRF is adapted from [40]. See the supplementary for the algorithm description.

where \tilde{x} comes from a mixture distribution $\tilde{\mu}$, so it can be replaced by a combination of x_t^r and x_t^g . η and γ are adjustable hyperparameters, to keep the penalty term positive, they are subject to $\gamma \geq -\eta \|\epsilon_{x_t^r; y, t} - \epsilon_{x_t^r; \phi, t}\|_2^2 / \|\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; \phi, t}\|_2^2$. In practice, we find $\eta = 1/2$, $\gamma \in [-1, 0)$ works for most cases. See supplementary materials for the derivation details of Eq. (9). In SDS [44], this discriminator optimization is completely ignored, as there is no optimizable ϕ . VSD [56] uses the LoRA or simple UNet branch to implement ϕ , and updates it using $\nabla_\phi \|\epsilon_{x_t^g; \phi, t} - \epsilon\|_2^2$, which is just a part of Eq. (9).

We propose Adversarial Score Distillation (ASD) based on the WGAN paradigm. Figure 3 illustrates the workflow of ASD. ASD uses Eq. (7) to update the parameters θ of a generator. This generator could be a NeRF [40, 41] in the text-to-3d task, or a general generator similar to that in [36], or just image pixels in the 2D distillation and image editing tasks. ASD maintains an optimizable discriminator by implementing ϕ with an optimizable conditional embedding [12] or LoRA [20], and updates this discriminator using the complete \mathcal{L}_D . However, in Eq. (9), the noisy real sample x_t^r is unavailable for text-conditioned cases. For these cases, we can use the upper bound of \mathcal{L}_D , which does not contain x_t^r terms.

\mathcal{L}'_D for text-conditioned distillation. When $\eta = 1/2$, based on triangle and Cauchy–Schwarz inequality, we have $\frac{1}{2} \|\epsilon_{x_t^r; y, t} - \epsilon_{x_t^r; \phi, t}\|_2^2 \leq \|\epsilon_{x_t^r; y, t} - \epsilon\|_2^2 + \|\epsilon_{x_t^r; \phi, t} - \epsilon\|_2^2$. Thus, we can rewrite Eq. (9) as

$$\begin{aligned} \nabla_\phi \mathcal{L}_D &\leq \nabla_\phi \mathbb{E}_{t, \epsilon} [\|\epsilon_{x_t^g; \phi, t} - \epsilon\|_2^2 \\ &\quad + \gamma \|\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; \phi, t}\|_2^2] := \nabla_\phi \mathcal{L}'_D \end{aligned} \quad (10)$$

For the case where real samples are unavailable, we use Eq. (10) to update the prompt ϕ of ASD. For cases where real samples are available, like image editing or image / 3D-conditioned distillation, we can continue to use Eq. (9).

\mathcal{L}_G^e for image editing. We extend ASD to image editing

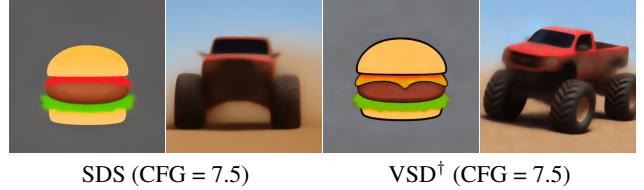


Figure 4. 2D score distillation results with the prompts “hamburger” and “a monster truck”. VSD^\dagger denotes that the LoRA branch is updated with 50 steps per iteration, resulting in over-smoothing images similar to SDS.

by using two discriminators with source prompts z and target prompts y . The new discriminator function is defined as $F(x_t; y, z) = D(x_t; y) - D(x_t; z)$, which gives high confidence for samples closing to the target prompt y but away from the source prompts z . Then, the generator loss will be

$$\begin{aligned} \nabla_\theta \mathcal{L}_G^e &= \nabla_\theta \mathbb{E}_{t, \epsilon} [\log p(\phi_y | x_t^g) - \lambda \log p(y | x_t^g) \\ &\quad - \log p(\phi_z | x_t^g) + \lambda \log p(z | x_t^g)] \\ &= \mathbb{E}_{t, \epsilon} [\omega(t) (\lambda (\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; t}) - (\epsilon_{x_t^g; \phi_y, t} - \epsilon_{x_t^g; t})) \\ &\quad - \lambda (\epsilon_{x_t^g; z, t} - \epsilon_{x_t^g; t}) + (\epsilon_{x_t^g; \phi_z, t} - \epsilon_{x_t^g; t}) \frac{\partial x_0^g}{\partial \theta}] \end{aligned} \quad (11)$$

For discriminator optimization, we can use either Eq. (9) or Eq. (10) in the image editing task. We find that DDS [17] is a special case of our paradigm. Specifically, DDS employs two fixed discriminators (similar to SDS), and feeds the noisy source image into the second discriminator $D(x_t; z)$.

5. Comparison and Discussion

Score Distillation Sampling. In practice, the SDS gradient comes from the generator loss in WGAN instead of the L2 diffusion loss. It maintains a fixed sub-optimal discriminator by using ϵ to approximate $\epsilon_{x_t^g; \phi, t}$, which ignores the inductive bias of the pretrained diffusion model. In contrast, ASD uses an optimizable conditional embedding or LoRA to implement ϕ , thus keeping the inductive bias in $\epsilon_{x_t^g; \phi, t}$.

Variational Score Distillation. From the methodology perspective, we employ the WGAN paradigm to explain and analyze VSD instead of the Wasserstein gradient flow, which is more straightforward. Based on the Wasserstein gradient flow [56], for each iteration, VSD needs to train the score function of the particle distribution (*i.e.*, the LoRA branch) until convergence. However, some studies [37, 53] point out that WGAN may fail when approximating the very accurate Wasserstein distance, which means that updating the LoRA branch with too many steps per iteration may degrade the results. Figure 4 shows the results if we update the LoRA branch with 50 steps per iteration. In addition, the discriminator optimization of VSD is just a part of the WGAN discriminator loss, which can be considered as a special case of ASD when setting γ in Eq. (10) to 0.



SDS (CFG = 10k) Only classifier term
(a) 2D score distillation results.



(b) 3D results by using the classifier term only.



(c) ASD recovers from over-saturated initialization.

Figure 5. Score distillation results with the prompt “a photograph of a fox”, “a front view of an owl” in 2D, and “a colorful rooster”, “a plush dragon toy” in 3D. Only using the classifier term is equivalent to using SDS with a huge CFG scale, which tends to get over-saturated results in both 2D and 3D distillation. ASD can recover from over-saturation initialization.

Classifier Score Distillation. Very recently, the concurrent work Classifier Score Distillation (CSD) [62] highlights the importance of the classifier term in the SDS gradient. We observe that only using the classifier term is similar to using SDS with a very huge CFG scale (as the learning rate can be adjusted dynamically by the optimizer, the percentage of the gradient component is the key factor), thereby prone to over-saturation in both 2D and 3D distillation. As a very important trick, CSD proposes to use negative prompts to alleviate over-saturation, which is similar to the discriminator in Eq (5) with fixed ϕ . Thanks to the adversarial capabilities, ASD can recover from over-saturation even with the over-saturated initialization. See Figure 5 for visualization.

Disscussion of γ . When $\gamma = 0$, Eq (10) becomes VSD, which can be considered as the WGAN [3, 58] discriminator loss with fixed penalty term for Lipschitz constraint. For $\gamma < 0$, we decrease the weight of the penalty term, which can enhance the ability of the discriminator. For $\gamma > 0$, we



a bear wearing sunglasses and a tie looks very proud



a flamingo in the city



a lively magical town inspired by Victorian England and Amsterdam

(a) Comparisons of 2D score distillation.



Sydney opera house, aerial view



a steaming hot plate piled high with spaghetti and meatballs



a small saguaro cactus planted in a clay pot

(b) Comparisons of text-conditioned 3D NeRF distillation.

Figure 6. Quality comparisons of distillation results among SDS, VSD, and ASD. For the text-to-3D task, the effects of different representations vary widely, so we all compared the results from NeRF distillation for fair play. “@7.5” denotes “at the CFG scale 7.5”. “T2I” denotes text-to-image results.

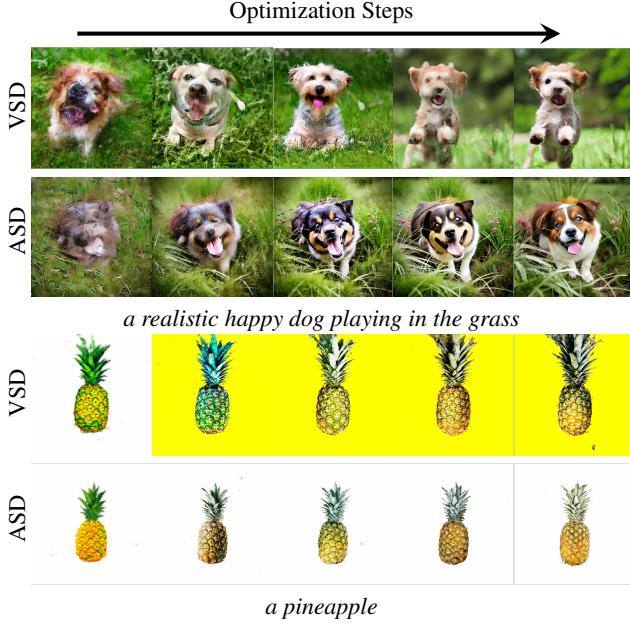


Figure 7. Stability comparisons of the distillation progress between VSD and SDS in both 2D and 3D. As can be seen, ASD constantly keeps the main object and image / NeRF structures unchanged.

increase the weight of the penalty term. Specifically, when $\gamma \gg 0$, the penalty term will dominate Eq (10), which leads to $\epsilon_{x_t^g, y, t} \approx \epsilon_{x_t^g, \phi, t}$. Then the discriminator fails to recognize fake samples, and the gradient for generators (6) mainly comes from the implicit classifier term. Only using the classifier term can be considered as a special case of our paradigm in this situation.

6. Experiments

Implementation Details. We choose ProlificDreamer-2D [65] as the code base for 2D score distillation, Three-studio [15] as the code base for the text-to-3D task, and DDS [17] as the code base for image editing. We employ Stable Diffusion v2-1-base [52] as the pretrained diffusion model except for the text-to-3D task, where the LoRA branch is initialized from Stable Diffusion v2-1 [46, 51], following the configuration of VSD [56] in Three-studio.

We set λ in Eq. (7) to 7.5 by default and use Eq. (10) for discriminator optimization in both 2D score distillation and text-to-3D tasks. We set γ in Eq. (10) to -1 for 2D score distillation, and set to -0.5 for the text-to-3D task. Although here we use fixed γ for simplicity, we think that a dynamic γ based on prompts and optimization steps may yield better results. In the text-to-3D task, all other configurations are inherited from the default configuration of VSD in Three-studio [15].

For image editing, we feed the source image into the sec-



Figure 8. Diversity evaluation. ASD can easily distill different 2D and 3D results by switching the random seed.

ond discriminator $D(x_t; z)$ following DDS [17]. In such case, we use Eq. (10) to optimize the first discriminator $D(x_t; y)$ and employ Eq. (9) to optimize the second discriminator $D(x_t; z)$, where γ in Eq. (10) is set to -1, η and γ in Eq. (9) are set to 0 and 1, respectively. The source image of image editing is generated from Stable Diffusion v2-1 [51].

Quality. We first compare the distillation quality among SDS [44], VSD [56], and ASD. Figure 6 shows the visual results obtained by different distillation methods in 2D distillation and text-to-3D tasks. In 2D distillation, Figure 6 (a) demonstrates that ASD can consistently achieve better results than SDS and VSD for three types of images: portraits, realistic images, and caricatures. In comparison, VSD results are slightly more saturated (as shown in the bear image) and contain artifacts due to instability in the distillation process (as shown in the flamingo image). Both phenomena are also seen in the text-to-3D task, where the VSD results are slightly more saturated in color and have more structural artifacts. Thanks to the complete discriminator optimization, the proposed ASD can obtain clean 3D NeRF compared to VSD.

Stability. We further evaluate the stability of ASD by comparing the distillation process with VSD. Figure 7 shows the distillation progress in 2D and 3D tasks. In 2D distillation, ASD always keeps the main objects and image structures unchanged. The gradient of the ASD generator loss is focused on color and details, where the color of the dog gradually changes from black to yellow. However, VSD is not that stable. Both the main object, the image structure,

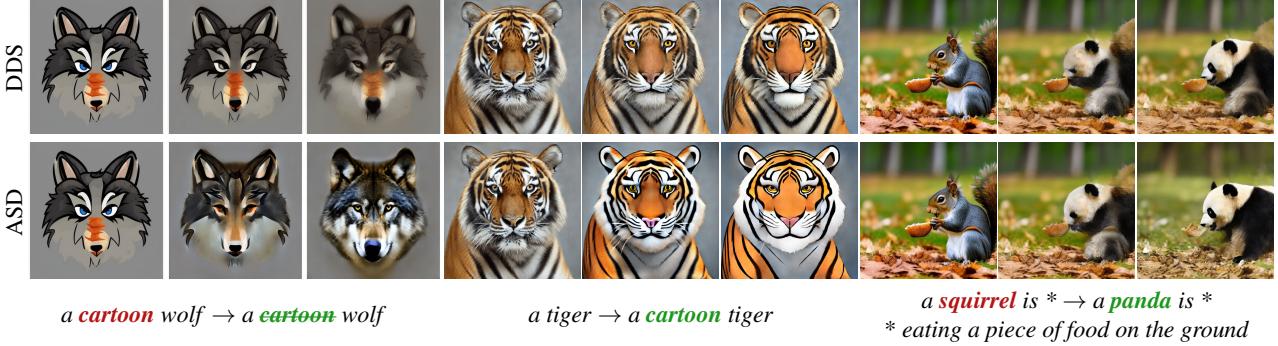


Figure 9. Comparisons between DDS and ASD on 2D image editing tasks. We test our ASD in three image editing situations: simplification, refinement, and replacement. For each case, we use three images to show the optimization progress.

the color of the dog, and other details are constantly changing during the VSD distillation process. We note that this instability has less impact on the text-to-3D task, as occupancy loss and NeRF help maintain structural stability. Despite this, the structures distilled by VSD still change faster than those distilled by ASD, resulting in structural artifacts (see the pineapple leaves). It is worth noting that both ASD and VSD will change objects or structures eventually, due to their adversarial nature. ASD can slow down and stabilize this process, thereby reducing artifacts.

Diversity. By switching seeds, we can easily get various distillation results. Figure 8 exhibits several distillation results using the same text prompt but different seeds. It can be seen that ASD can always obtain high-quality results in both 2D and 3D tasks.

Image Editing. In addition to typical score distillation tasks, we further extend ASD to zero-shot image editing. Figure 9 shows how ASD simplifies the source image, refines the source image, and replaces objects in the source image. For the simplification tasks, our ASD can remove the most unnecessary high-frequency details and retain the main content based on the target prompt. ASD can also add realistic details back to cartoon-style images, which is significantly better than the recent zero-shot image editing method DDS [17]. We also challenged the object replacement task. Our ASD method produces more reasonable, detailed, and realistic results based on semantic information instead of the rigid fashion of copying fixed objects and then pasting them at fixed locations. See Figure 9, the squirrel’s tail still appears in the results of DDS. Besides reasonable object replacement, ASD can maintain the details of the source image as well.

Quantitative Results. Table 1 presents quantitative evaluations of CLIP scores and user studies for 3D generation tasks, demonstrating that ASD performs favorably against SDS and VSD. See the supplementary for more discussion about CLIP scores.

Method	CLIP Score		
	CLIP B/32↑	CLIP B/16↑	CLIP L/14↑
DreamFusion (SDS)	0.3282	0.3290	0.2889
ProlificDreamer (VSD)	0.3203	0.3258	0.2789
Ours (ASD)	0.3314	0.3376	0.2854
User Study			
Method	Details(%)	Semantic Alignment (%)	Overall Quality (%)
ProlificDreamer (VSD)	35.3	45.7	34.5
Ours (ASD)	64.7	54.3	65.5

Table 1. Quantitative Results. CLIP scores are the cosine similarity between multiple randomly rendered views of the 3D model and the given text prompt. User evaluation is conducted between VSD and ASD from three perspectives.

7. Limitation

Despite showing favorable capabilities in terms of distillation quality, stability, and diversity, the speed of ASD is similar to VSD. We find that sharing the same added noise between generator optimization and discriminator optimization can speed up distillation by 30 ~ 50%, with a little quality sacrifice. However, distilling a 3D NeRF still takes more than 5 hours on a single A800 GPU. Using faster 3D representation [26], or faster PEFT [11, 20, 31, 33, 35] methods to update the discriminator, or designing forward-only methods to approximate $\epsilon_{xt;\phi,t}$ may be promising solutions to this limitation.

8. Conclusion

In this paper, we proposed the Adversarial Score Distillation (ASD) under the WGAN paradigm. Compared to existing methods, our ASD exploits a complete discriminator loss with a principled discussion, which alleviates the CFG-sensitive issue and leads to impressive performances.

More importantly, we elucidated the relationship between score distillation and GAN, which not only explains methodological issues of existing score distillation, but also enables the pretrained model to extend to downstream tasks by various GAN paradigms.

9. Acknowledgements

We would like to express our gratitude to all the individuals who provided assistance in this work: Chao-hui Yu, Fan Wang. We also thank the authors of all the open-source codes and models that we used.

References

- [1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. [3](#)
- [2] Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. [3](#)
- [3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. [2, 3, 4, 6, 1](#)
- [4] Mohammadreza Armandpour, Huangjie Zheng, Ali Sadeghian, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023. [1](#)
- [5] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018. [3](#)
- [6] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. [3](#)
- [7] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22246–22256, 2023. [1](#)
- [8] Yiwen Chen, Chi Zhang, Xiaofeng Yang, Zhongang Cai, Gang Yu, Lei Yang, and Guosheng Lin. It3d: Improved text-to-3d generation with explicit view synthesis. *arXiv preprint arXiv:2308.11473*, 2023. [1](#)
- [9] Yunje Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. [3](#)
- [10] Brian Dolhansky and Cristian Canton Ferrer. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7902–7911, 2018. [3](#)
- [11] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. Krona: Parameter efficient tuning with kronecker adapter. *arXiv preprint arXiv:2212.10650*, 2022. [8](#)
- [12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*, 2022. [3, 4, 5, 2](#)
- [13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. [2, 3, 4](#)
- [14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. [3](#)
- [15] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. <https://github.com/threestudio-project/threestudio>, 2023. [7](#)
- [16] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019. [3](#)
- [17] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2328–2337, 2023. [1, 3, 5, 7, 8](#)
- [18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. [1, 4](#)
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [2](#)
- [20] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. [3, 4, 5, 8, 2](#)
- [21] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. [3](#)
- [22] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017. [3](#)
- [23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. [3](#)
- [24] Abdul Jabbar, Xi Li, and Bourahla Omar. A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8):1–49, 2021. [3](#)
- [25] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. [1](#)

- [26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 8
- [27] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. Collaborative score distillation for consistent visual synthesis. *arXiv preprint arXiv:2307.04787*, 2023. 1
- [28] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. PMLR, 2017. 3
- [29] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. 2
- [30] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018. 3
- [31] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021. 8
- [32] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 1
- [33] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022. 8
- [34] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017. 3
- [35] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 2023. 8
- [36] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 5
- [37] Anton Mallasto, Guido Montúfar, and Augusto Gerolin. How well do wgan estimate the wasserstein metric? *arXiv preprint arXiv:1910.03875*, 2019. 5
- [38] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3
- [39] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 1
- [40] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 5
- [41] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 5
- [42] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. *Advances in neural information processing systems*, 31, 2018. 3
- [43] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d {gan}s know 3d shape? unsupervised 3d shape reconstruction from 2d image {gan}s. In *International Conference on Learning Representations*, 2021. 3
- [44] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations*, 2022. 1, 3, 5, 7
- [45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1
- [46] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 7
- [47] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 1
- [48] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International Conference on Learning Representations*, 2023. 1
- [49] Edward J Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In *Conference on Robot Learning*, pages 87–96. PMLR, 2017. 3
- [50] Kunpeng Song, Ligong Han, Bingchen Liu, Dimitris Metaxas, and Ahmed Elgammal. Diffusion guided domain adaptation of image generators. *arXiv preprint arXiv:2212.04473*, 2022. 3
- [51] StabilityAI. Stable diffusion 2-1. <https://huggingface.co/stabilityai/stable-diffusion-2-1>, 2023. 2, 7
- [52] StabilityAI. Stable diffusion 2-1-base. <https://huggingface.co/stabilityai/stable-diffusion-2-1-base>, 2023. 2, 7
- [53] Jan Stanczuk, Christian Etmann, Lisa Maria Kreusser, and Carola-Bibiane Schönlieb. Wasserstein gans work because

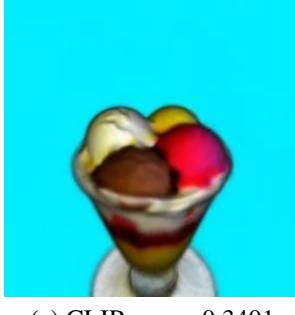
- they fail (to approximate the wasserstein distance). *arXiv preprint arXiv:2103.01678*, 2021. 5
- [54] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing*, 27(8):4066–4079, 2018. 3
 - [55] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 1, 3
 - [56] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3, 4, 5, 7
 - [57] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 3
 - [58] Jiqing Wu, Zhiwu Huang, Janine Thoma, Dinesh Acharya, and Luc Van Gool. Wasserstein divergence for gans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 4, 6, 1
 - [59] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2017. 3
 - [60] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dual-gan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017. 3
 - [61] Chaohui Yu, Qiang Zhou, Jingliang Li, Zhe Zhang, Zhibin Wang, and Fan Wang. Points-to-3d: Bridging the gap between sparse points and shape-controllable text-to-3d generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6841–6850, 2023. 1
 - [62] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023. 1, 6
 - [63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3
 - [64] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold, 2018. 3
 - [65] Yuanzhi Zhu. Prolific dreamer 2d. https://github.com/yuanzhi-zhu/prolific_dreamer2d, 2023. 7

A. Quantitative Evaluations and User Studies for 3D Generation

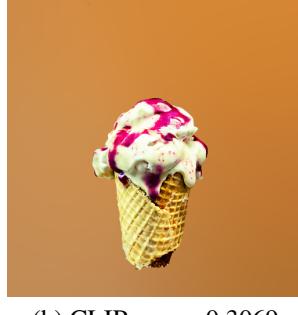
We quantitatively compare our ASD with DreamFusion [44] (SDS) and ProlificDreamer [56] (VSD) using CLIP score, following previous works [25, 44]. We evaluate the similarities between text prompts and randomly rendered views of the 3D NeRFs using three variants of CLIP [45] (*i.e.*, CLIP B/32, CLIP B/16, and CLIP B/14). The CLIP score may not honestly reflect the quality of 3D results. For example, the CLIP score of Figure 10(a) is higher than Figure 10(b) even though its quality is significantly worse, which does not align with human perceptual judgments. Several existing methods [7, 56] choose to only conduct user studies for performance evaluation.

Method	CLIP B/32↑	CLIP B/16↑	CLIP L/14↑
DreamFusion (SDS)	0.3282	0.3290	0.2889
ProlificDreamer (VSD)	0.3203	0.3258	0.2789
Ours (ASD)	0.3314	0.3376	0.2854

Table 2. CLIP Score by computing the similarity between multiple randomly rendered views of the 3D model and the given text prompt.



(a) CLIP score: 0.3401



(b) CLIP score: 0.3069

Figure 10. Illustration of CLIP scores that conflict with human perceptual judgments. We use CLIP B/16 to compute scores here.

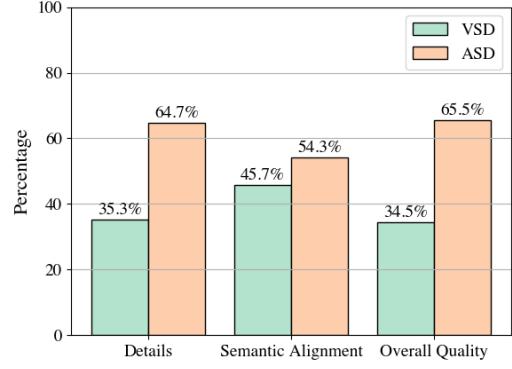


Figure 11. User evaluation between VSD and ASD.

Therefore, we also present user studies in Figure 11. We compare 3D NeRFs generated by VSD and ASD with different prompts in terms of details, semantic alignment, and overall quality. Figure 11 demonstrates that ASD leads to better details, which is consistent with our observation that VSD’s results are slightly more saturated and therefore have worse details. The semantic alignment scores of VSD and ASD are similar. In general, more users prefer the results generated by ASD.

B. Derivation of Adversarial Score Distillation Loss and Gradients

We define a discriminator in the form as

$$D(x_t; y) = \log \frac{p(y|x_t)}{p(\phi|x_t)} \quad (12)$$

Based on WGAN [3, 58], we have the generator loss as

$$\mathcal{L}_G = \mathbb{E}_{t,\epsilon}[-D(x_t^g; y)] = \mathbb{E}_{t,\epsilon}[-\log \frac{p(y|x_t)}{p(\phi|x_t)}] \leq \mathbb{E}_{t,\epsilon}[-\log \frac{p(y|x_t)}{p(\phi|x_t)}^\lambda] := \mathcal{L}'_G \quad (13)$$

where $x_t^g = \alpha_t g(\theta, c) + \sigma_t \epsilon$ denotes the generated sample with noise. According to the Bayesian rule, we can do the transformation $\nabla_{x_t} \log p(y|x_t) = -1/\sigma_t (\epsilon_{x_t; y, t} - \epsilon_{x_t, t})$ similar to CFG [18]. Thus, we have

$$\begin{aligned} \nabla_\theta \mathcal{L}'_G &= \nabla_\theta \mathbb{E}_{t,\epsilon}[\log p(\phi|x_t^g) - \lambda \log p(y|x_t^g)] \\ &= \mathbb{E}_{t,\epsilon}[\omega(t)(-\left(\epsilon_{x_t^g, \phi, t} - \epsilon_{x_t^g, t}\right) + \lambda(\epsilon_{x_t^g, y, t} - \epsilon_{x_t^g, t})) \frac{\partial x_0^g}{\partial \theta}] \\ &= \mathbb{E}_{t,\epsilon}[\omega(t)(\epsilon_{x_t^g, t} + \lambda(\epsilon_{x_t^g, y, t} - \epsilon_{x_t^g, t}) - \epsilon_{x_t^g, \phi, t}) \frac{\partial x_0^g}{\partial \theta}] \end{aligned} \quad (14)$$

where $\omega(t)$ is a weight based on the time t .

Then we have the discriminator loss as

$$\begin{aligned}\mathcal{L}_D &= \mathbb{E}_{t,\epsilon}[D(x_t^g; y) - D(x_t^r; y) + \tau \|\nabla_{\tilde{x}_t} D(\tilde{x}_t; y)\|_2^2] \\ &= \mathbb{E}_{t,\epsilon}[\log \frac{p(y|x_t^g)}{p(\phi|x_t^g)} - \log \frac{p(y|x_t^r)}{p(\phi|x_t^r)} + \tau \|\nabla_{\tilde{x}_t} \log \frac{p(y|\tilde{x}_t)}{p(\phi|\tilde{x}_t)}\|_2^2]\end{aligned}\quad (15)$$

where $x_t^r = \alpha_t x_0^r + \sigma_t \epsilon$ is the noisy real sample based on the real sample x_0^r and the time t . In Eq. (15), $p(y|x_t^r)$ and $p(y|x_t^g)$ are based on the given pretrained diffusion model, so they are not optimizable. ϕ in $p(\phi|x_t^g)$ and $p(\phi|x_t^r)$ is optimizable, which can be implemented using an textual-inversion embedding [12] or LoRA [20]. Thus, \mathcal{L}_D is a loss function only about parameters ϕ , its gradient can be derived as

$$\begin{aligned}\nabla_\phi \mathcal{L}_D &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\log p(\phi|x_t^r) - \log p(\phi|x_t^g) + \tau \|\nabla_{\tilde{x}_t} \log \frac{p(y|\tilde{x}_t)}{p(\phi|\tilde{x}_t)}\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\left(\log p(x_t^r|\phi) + \log p(\phi) - \log p(x_t^r)\right) - \left(\log p(x_t^g|\phi) + \log p(\phi) - \log p(x_t^g)\right) + \tau \|\nabla_{\tilde{x}_t} \log \frac{p(y|\tilde{x}_t)}{p(\phi|\tilde{x}_t)}\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\log \frac{p(x_t^r|\phi)}{p(x_t^g|\phi)} + \tau \|\nabla_{\tilde{x}_t} (\left(\log p(\tilde{x}_t|y) + \log p(y) - \log p(\tilde{x}_t)\right) - \left(\log p(\tilde{x}_t|\phi) + \log p(\phi) - \log p(\tilde{x}_t)\right))\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\log p(x_t^r|\phi) - \log p(x_t^g|\phi) + \tau \|\nabla_{\tilde{x}_t} \log p(\tilde{x}_t|y) - \nabla_{\tilde{x}_t} \log p(\tilde{x}_t|\phi)\|_2^2] \\ &\approx \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \eta \|\epsilon_{x_t^r,y,t} - \epsilon_{x_t^r,\phi,t}\|_2^2 + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2]\end{aligned}\quad (16)$$

where we use $-\log p(x_0) \approx \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t,\phi,t} - \epsilon\|_2^2]$, according to [19, 29]. \tilde{x} comes from a mixture distribution $\tilde{\mu}$, so we can sampling both real or fake data points [58] which means \tilde{x}_t can be replaced by a combination of x_t^r and x_t^g . η and γ are adjustable hyperparameters, to keep the penalty term positive, they are subject to $\gamma \geq -\eta \|\epsilon_{x_t^r,y,t} - \epsilon_{x_t^r,\phi,t}\|_2^2 / \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2$. However, in Eq. (16), the noisy real sample x_t^r is unavailable for text-conditioned cases. For these cases, we can use the upper bound of \mathcal{L}_D , (i.e. \mathcal{L}'_D), which does not contain x_t^r terms. When $\eta = 1/2$, based on triangle and Cauchy–Schwarz inequality, we have:

$$\begin{aligned}\nabla_\phi \mathcal{L}'_D &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + (\|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \|\epsilon_{x_t^r,y,t} - \epsilon\|_2^2) + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \frac{1}{2}(1^2 + 1^2)(\|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \|\epsilon_{x_t^r,y,t} - \epsilon\|_2^2) + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2] \\ &\geq \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \frac{1}{2}(\|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2 + \|\epsilon_{x_t^r,y,t} - \epsilon\|_2)^2 + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2] \\ &\geq \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \frac{1}{2}(\|\epsilon_{x_t^r,\phi,t} - \epsilon + \epsilon - \epsilon_{x_t^r,y,t}\|_2)^2 + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2] \\ &= \nabla_\phi \mathbb{E}_{t,\epsilon}[\|\epsilon_{x_t^g,\phi,t} - \epsilon\|_2^2 - \|\epsilon_{x_t^r,\phi,t} - \epsilon\|_2^2 + \frac{1}{2}\|\epsilon_{x_t^r,y,t} - \epsilon_{x_t^r,\phi,t}\|_2^2 + \gamma \|\epsilon_{x_t^g,y,t} - \epsilon_{x_t^g,\phi,t}\|_2^2] = \nabla_\phi \mathcal{L}_D\end{aligned}\quad (17)$$

In practice, we find $\eta = 1/2$, $\gamma \in [-1, 0)$ works for most cases.

C. Explanation of the workflow

To get a clearer picture of the optimization workflow, we further provide an algorithm description in Algorithm 1.

D. Multi-view visualization

We exhibit some examples in Figure 12. The code and project page will be released which will contain rotating videos of the generated 3D models.

E. Additional Experimental Results

More examples in both 2D distillation and text-to-3D tasks are shown in Figure 13. In the text-to-3D task, we only generate 3D NeRFs from scratch for simplicity, which is the first stage of VSD.

Algorithm 1 Adversarial Score Distillation for 3D tasks

Input: Prompt y , optimizable textual-inversion embedding or LoRA ϕ , and pretrained diffusion model.

- 1: **repeat**
- 2: Sample a camera pose c .
- 3: Differentiable render the 3D structure θ at pose c to get a 2D image $x_0 = g(\theta, c)$.
- 4: Random select a timestep t and add random noise ϵ to get $x_t^g = \alpha_t x_0 + \sigma_t \epsilon$.
- 5: Predict noise $\epsilon_{x_t^g; \phi, t}$, $\epsilon_{x_t^g; y, t}$ and $\epsilon_{x_t^g; t}$ using the pretrained diffusion model.
- 6: (Update G) Update θ with the gradient $\nabla_\theta \mathcal{L}'_G : \mathbb{E}_{t, \epsilon} [\omega(t) (\epsilon_{x_t^g; t} + \lambda (\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; t}) - \epsilon_{x_t^g; \phi, t}) \frac{\partial x_0^g}{\partial \theta}]$
- 7: (Update D) Update ϕ with the gradient $\nabla_\phi \mathcal{L}'_D : \nabla_\phi \mathbb{E}_{t, \epsilon} [\|\epsilon_{x_t^g; \phi, t} - \epsilon\|_2^2 + \gamma \|\epsilon_{x_t^g; y, t} - \epsilon_{x_t^g; \phi, t}\|_2^2]$
- 8: **until** converged

Output: 3D structure θ .



Figure 12. Multi-view visualizations of generated 3D results.



a castle in the middle of
a marsh



a boy portrait with
sunglasses



a cozy living room with a
painting of a corgi [...]



a group of elephants walking
in muddy water



a photograph of a hamster



a red fire hydrant spraying
water



a small kitchen with a low
ceiling



cliffs at day time



a tarantula, highly detailed



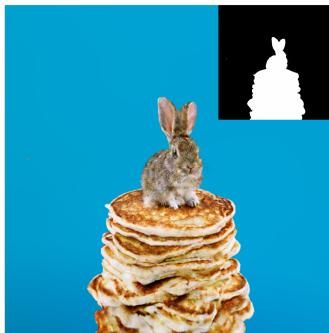
a delicious croissant



a dachshund dressed up [...]



[...] chocolate chip cookies



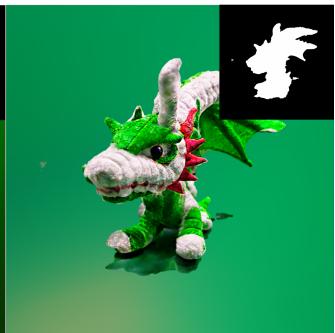
a baby bunny sitting on [...]



a blue jay standing on [...]



[...] a goose made out of gold



a plush dragon toy

Figure 13. More examples generated by ASD in both 2D and 3D.